

System Automation mit Puppet und Foreman

Toni Schmidbauer

27. März 2014

whoami

- ▶ SysAdmin@s-itsolutions
- ▶ toni@stderr.at
- ▶ <http://github.com/tosmi>
- ▶ stderr@jabber.org

Agenda

- ▶ Kurze Umfrage
- ▶ Was ist Puppet?
- ▶ Was ist Foreman?
- ▶ Puppet@s-iTSolutions
- ▶ Was haben wir geplant

Umfrage

Was ist Puppet?

```
1  class linuxwochen2014 {  
2  
3      user { 'linuxwochen':  
4          ensure => present,  
5          uid => 4711,  
6          gid => 4711,  
7      }  
8  
9      package { 'emacs':  
10         ensure => installed  
11     } ->  
12     package { 'vi':  
13         ensure => absent,  
14     }  
15 }
```

Zuordnung von Klassen

```
1 node linuxwochen {  
2     include mypuppetconfig  
3 }
```

- ▶ oder über einen External Node Classifier (Foreman)

Was ist Foreman?

 FOREMAN

Dashboard Hosts Reports Facts Audits Statistics Trends More ▾

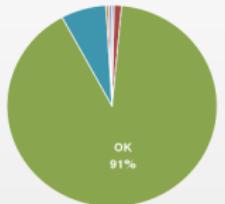
Overview

Filter ... ▾

Host Configuration Status

Hosts that had performed modifications without error	3
Hosts in error state	11
Good host reports in the last 35 minutes	875
Hosts that had pending changes	0
Out of sync Hosts	67
Hosts with no reports	4
Hosts with alerts disabled	5

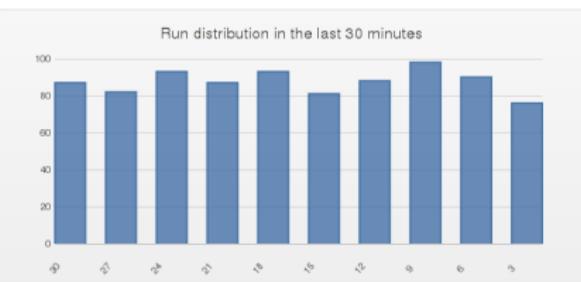
Total Hosts: 965



Latest Events

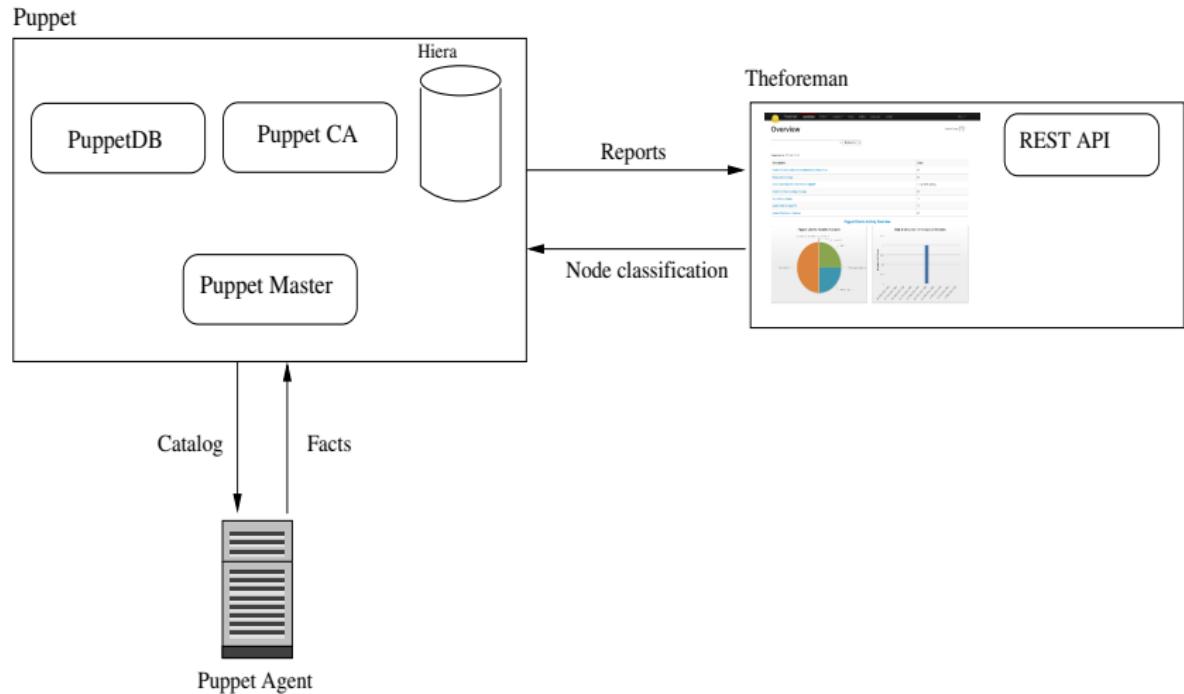
Host	A	R	F	FR	S	P
appplusu2.l...	0	0	0	0	8	0
appqsys2.se...	0	0	0	0	8	0
appplusu2.i...	0	0	0	0	8	0
appqsys2.se...	0	0	0	0	8	0
appplusu2.l...	0	0	0	0	8	0
appqsys2.se...	0	0	0	0	8	0

Run distribution in the last 30 minutes





Puppet run



Und jetzt?



- ▶ Wie verwaltet wir unseren Puppet Code?
- ▶ Wie soll unsere Puppet Umgebung aussehen?
- ▶ Wie erfolgt das Deployment des Codes?
- ▶ Wie organisieren wir Module?
- ▶ Wie soll eine Entwicklungsumgebung aussehen?
- ▶ Wie testen wir den Puppet Code?
- ▶ Wie verwalten wir Module von PuppetForge?

Wie verwaltet wir unseren Puppet
Code?

Wie organisieren wir Module?

GIT

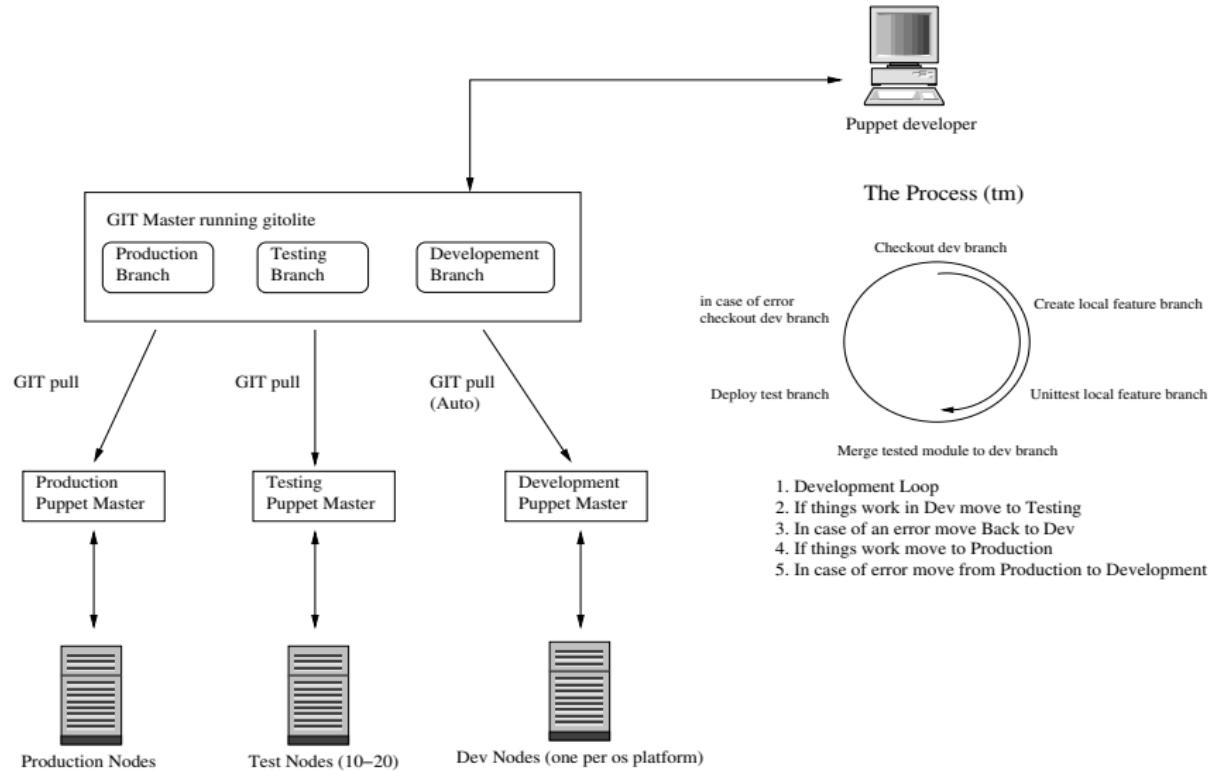
- ▶ Ein zentrales GIT Repository
- ▶ Berechtigungssystem für GIT mit gitolite
- ▶ 3 Hauptbranches
 - ▶ Master: Staging via GIT pull auf 4 Dev Server
 - ▶ Testing: ca. 25 "Produktions" Server (git pull)
 - ▶ Production: der Rest, Staging via tags
- ▶ Feature Branches für neue Module

Demo

Wie soll unsere Puppet Umgebung aussehen?

Wie erfolgt das Deployment des Codes?

Puppet Umgebung



Puppet Umgebung II

- ▶ Am Development Master verwenden wir dynamische Puppet Environments

<http://puppetlabs.com/blog/git-workflow-and-puppet-environments>

- ▶ Jeder Feature Branch wird ein eigenes Puppet Environment
- ▶ Das Staging in die Produktion erfolgt über GIT tags

Deployment

Jenkins

Search | Help | Log out | Disable Auto Refresh

Build Pipeline: Puppet Development

Pipeline #2760 #2759 #2758 #2757 #2756

#2760 Puppet Development Rspec Tests (Mar 27, 2014 6:55:00 PM) 1 min 15 sec (Mar 27, 2014 6:55:15 PM) 1 min 17 sec (Mar 27, 2014 6:55:20 PM) 1 min 18 sec

#2759 Puppet Development Rspec Tests (Mar 27, 2014 6:55:00 PM) 1 min 15 sec (Mar 27, 2014 6:55:15 PM) 1 min 17 sec (Mar 27, 2014 6:55:20 PM) 1 min 18 sec

#2758 Puppet Development Rspec Tests (Mar 27, 2014 6:54:40 PM) 1 min 22 sec (Mar 27, 2014 6:55:00 PM) 1 min 17 sec (Mar 27, 2014 6:55:15 PM) 1 min 18 sec

#2757 Puppet Development Rspec Tests (Mar 27, 2014 6:54:40 PM) 1 min 17 sec (Mar 27, 2014 6:55:00 PM) 1 min 17 sec (Mar 27, 2014 6:55:15 PM) 1 min 18 sec

#2756 Puppet Development Rspec Tests (Mar 27, 2014 6:54:40 PM) 1 min 17 sec (Mar 27, 2014 6:55:00 PM) 1 min 17 sec (Mar 27, 2014 6:55:15 PM) 1 min 18 sec

Run History Configure Add Step Delete Merge

#25 Puppet Development List Check (Mar 27, 2014 6:55:00 PM) 1 min 7 sec

#15 Puppet Development Deployment (Mar 27, 2014 6:55:00 PM) 1 min 1 sec

#11 Puppet Development Update Foreman Classes (Mar 27, 2014 6:55:00 PM) 1 min 1 sec

#34 Puppet Development List Check (Mar 27, 2014 6:55:00 PM) 1 min 7 sec

#14 Puppet Development Deployment (Mar 27, 2014 6:55:00 PM) 1 min 1 sec

#10 Puppet Development Update Foreman Classes (Mar 27, 2014 6:55:00 PM) 1 min 1 sec

#33 Puppet Development List Check (Mar 27, 2014 6:55:00 PM) 1 min 7 sec

#13 Puppet Development Deployment (Mar 27, 2014 6:55:00 PM) 1 min 1 sec

#9 Puppet Development Update Foreman Classes (Mar 27, 2014 6:55:00 PM) 1 min 1 sec

#23 Puppet Development List Check (Mar 27, 2014 6:55:00 PM) 1 min 7 sec

#12 Puppet Development Deployment (Mar 27, 2014 6:55:00 PM) 1 min 1 sec

#8 Puppet Development Update Foreman Classes (Mar 27, 2014 6:55:00 PM) 1 min 1 sec

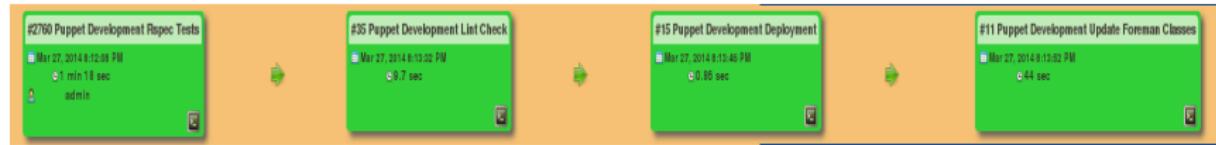
Puppet Development List Check (Mar 27, 2014 6:55:00 PM) 1 min 7 sec

Puppet Development Deployment (Mar 27, 2014 6:55:00 PM) 1 min 1 sec

Puppet Development Update Foreman Classes (Mar 27, 2014 6:55:00 PM) 1 min 1 sec

Help us broaden this page Page generated: Mar 27, 2014 6:25:12 PM | BEST MR | Jenkins ver. 1.532.2

Deployment



Monitoring

Monitor 

Puppet Development Deployment #15	0s	Puppet Development Lint Check #35	9s
Puppet Development Rspec Tests #2760	1m 19s	Puppet Development Update Foreman Classes #11	45s
Puppet Production #2722	43s	Puppet Testing Lint Check #2	16s
Puppet Testing Rspec Tests #2730	58s		

Possible Culprits: wce, toni, nenoscio

Brought to you by Just M&M

Wie soll eine Entwicklungsumgebung aussehen?



Vagrant

- ▶ <http://vagrantup.com>
- ▶ Ermöglicht virtuelle Entwicklungsumgebungen
- ▶ Vagrant Box ist ein vorkonfiguriertes Image
- ▶ Default VirtualBox andere Provider via Plugins (VMWare, KVM)

Demo

Wie testen wir den Puppet Code?

rspec-puppet

- ▶ Ruby RSpec Tests für Puppet
- ▶ Ist in unserer Vagrant Umgebung vorinstalliert
- ▶ Jedes Module muss RSpec Tests mitbringen
- ▶ Commit in Master Branch löst einen Jenkins Build aus
- ▶ Commit in Testing Branch löst einen Jenkins Build aus
- ▶ Commit in Production Branch löst einen Jenkins Build aus

Wie verwalten wir Module von PuppetForge?

Puppetforge Module

- ▶ Eigenes GIT Repository (puppetforge.git)
- ▶ Download der Module in der Entwicklungsumgebung
- ▶ Staging wie unser Haupt Puppet Repository

- ▶ Wie verwaltet wir unseren Puppet Code? *DONE*
- ▶ Wie soll unsere Puppet Umgebung aussehen? *DONE*
- ▶ Wie erfolgt das Deployment des Codes? *DONE*
- ▶ Wie organisieren wir Module? *DONE*
- ▶ Wie soll eine Entwicklungsumgebung aussehen? *DONE*
- ▶ Wie testen wir den Puppet Code? *DONE*
- ▶ Wie verwalten wir Module von PuppetForge? *DONE*



Probleme, Probleme, Probleme...

- ▶ Ein GIT Repo funktioniert nicht bei Änderungen von Upstream Modulen
- ▶ Andere Abteilungen sollen ihre Module unabhängig Testen
- ▶ Unitests sagen noch nichts aus wie sich der Code am Live-System verhält
- ▶ Wir sollten eigentlich das Zusammenspiel aller Modelle testen (Forge und eigene)

Was haben wir geplant?

- ▶ r10k für Deployment
- ▶ Ein Repository pro Module
- ▶ Puppet coverage tests
- ▶ Acceptance Tests mit Beaker